

2 BAB II

TINJAUAN PUSTAKA

2.1 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. (Sutoyo, et al., 2009, p. 9)

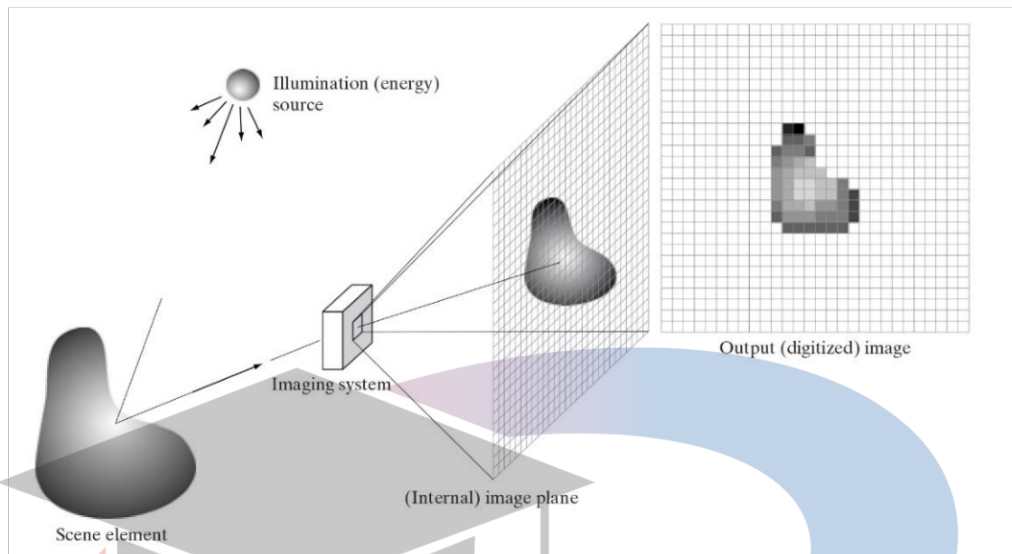
2.1.1 Citra Analog dan Citra Digital

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak dikertas foto, lukisan, pemandangan alam, hasil CT scan dan lain sebagainya. Citra analog tidak dapat dipresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog diantaranya adalah video kamera analog, kamera foto analog dan CT scan.

Sedangkan, citra digital adalah sebuah larik (*array*) yang berisi nilai - nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu. (Putra, 2010, p. 19).

2.1.2 Proses Akuisisi Citra

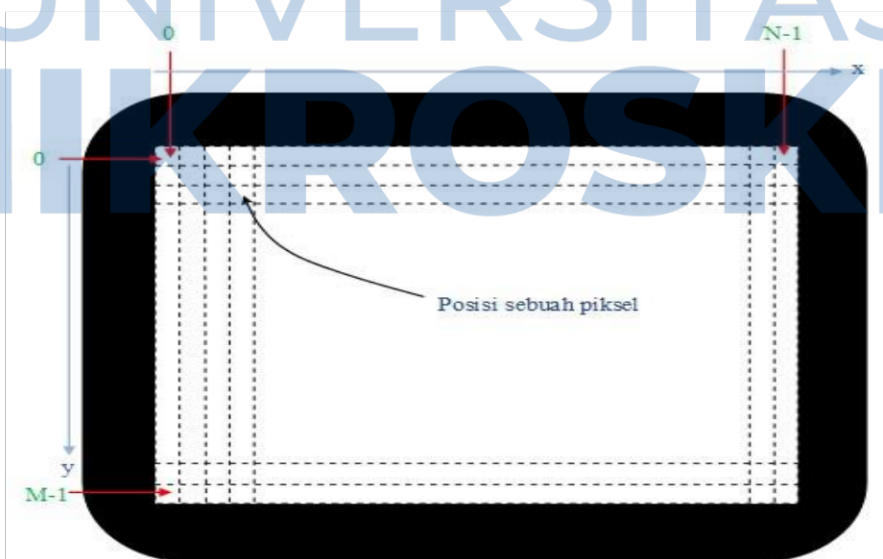
Proses akuisisi citra adalah pemetaan suatu pandangan (*scene*) menjadi citra kontinu dengan menggunakan sensor. Ada beberapa macam sensor untuk akuisisi citra, yaitu sensor tunggal (*single sensor*), sensor garis (*sensor strip*), dan sensor larik (*sensor array*). Proses akuisisi citra dapat dilihat pada gambar 2.1. (Putra, 2010, p. 28).



Gambar 2.1 Proses akuisisi citra digital
(Sumber: Gonzalez & Woods, 2002, p. 20).

2.1.3 Representasi Citra Digital

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau “picture element”). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan di Gambar 2.2.



Gambar 2.2 Sistem Koordinasi Citra digital

(Sumber: Kadir & Susanto, 2013, p. 10)

Citra digital yang berukuran $M \times N$ dapat ditulis dalam bentuk matrik sebagai berikut.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad \dots (2.1)$$

Nilai pada suatu irisan antara baris dan kolom pada posisi (x, y) disebut dengan *picture elements*, *image elements*, *pels*, atau *pixels*. Istilah terakhir (*pixel*) paling sering digunakan pada citra digital. (Putra, 2010, p. 20).

2.1.4 Jenis Citra

Nilai suatu piksel memiliki nilai dalam rentang tertentu dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya.

1. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra monokrom atau B&W (*black and white*). Hanya dibutuhkan 1 bit untuk mewakili nilai tiap pixel dari citra biner.

2. Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu kanal pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan dan putih.

3. Citra Warna (8 bit)

Setiap piksel dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna.

4. Citra Warna (16 bit)

Citra warna 16 bit (biasanya disebut sebagai citra *highcolor*) dengan setiap pikselnya diwakili dengan 2 *byte memory* (16 bit). Warna 16 bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau.

5. Citra Warna (24 bit)

Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualkan seluruh warna yang dapat dilihat penglihatan manusia. (Putra, 2010, pp. 39-44).

2.1.5 Format File Citra

Format *file* citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah *file*. Setiap format memiliki karakteristik masing-masing. Berikut adalah penjelasan beberapa format citra yang umum digunakan saat ini:

1. Bitmap (*.bmp)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai piksel.

2. JPEG (*.jpg)

Format .jpg adalah format yang sangat umum digunakan saat ini, khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

3. *Graphics Interchange Format* (*.gif)

Format ini dapat digunakan pada citra warna dengan palet 8 bit. Pada umumnya digunakan pada aplikasi web. Kualitas yang rendah

menyebabkan format ini tidak terlalu populer di kalangan peneliti pengolahan citra digital.

4. *Tagged Image Format* (*.tif, *.tiff)

Format ini merupakan format citra yang dapat digunakan untuk menyimpan citra *bitmap* hingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

5. *Portable Network Graphics* (*.png)

Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra *grayscale*, citra dengan palte warna dan juga citra *full color*. Format .png juga mampu menyimpan informasi hingga kanal *alpha* dengan penyimpanan sebesar 1 hingga 16 bit per kanal.

6. MPEG (*.mpg)

Format ini digunakan di dunia *internet* dan diperuntukkan sebagai format penyimpanan citra bergerak (*video*).

7. RGB (*.rgb)

Format ini merupakan format penyimpanan citra yang dibuat oleh *silicon graphics* untuk menyimpan citra berwarna.

8. RAS (*.ras)

Format .ras digunakan untuk menyimpan citra RGB tanpa kompresi.

9. *Postscript* (*.pas, *.epas, *.epfs)

Format ini diperkenalkan sebagai format untuk menyimpan citra buku elektronik. Dalam format ini, citra direpresentasikan ke dalam deret nilai desimal atau heksidesimal yang dikodekan ke dalam ASCII.

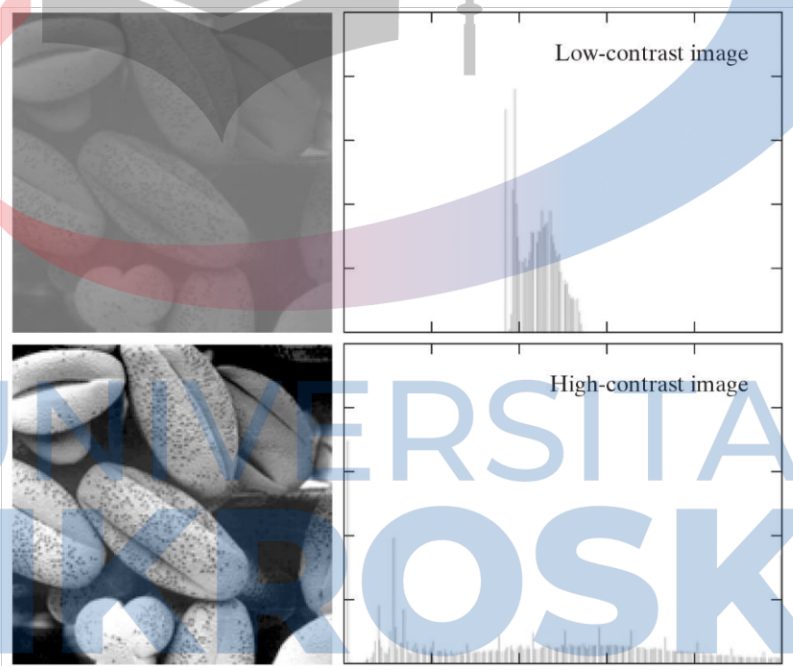
10. *Portable Image File Format*

Format ini memiliki beberapa bagian, di antaranya adalah *portable bitmap*, *portable graymap*, *portable pixmap*, dan *portable network map*, dengan format berturut-turut adalah .pbm, .pgm, .ppm, dan .pnm. (Putra, 2010)

2.1.6 Contrast, Low Contrast dan High Contrast.

Kontras suatu citra adalah distribusi atau tingkat penyebaran piksel-piksel ke dalam intensitas warna. Sebuah citra *grayscale* dengan kontras rendah maka akan terlihat terlalu gelap, terlalu terang, atau terlalu abu-abu. Histogram citra dengan kontras rendah, semua *pixel* akan terkonsentrasi pada sisi kiri, sisi kanan atau ditengah (Gambar 2.3). Semua *pixel* akan berkelompok secara rapat pada suatu sisi tertentu dan menggunakan sebagian kecil dari semua kemungkinan nilai *pixel*. (Putra, 2010)

Citra dengan kontras tinggi memiliki daerah gelap dan tereang yang luas. Histogram citra dengan kontras tinggi memiliki perataan yang merata di semua bagian histogram (Gambar 2.3).



Gambar 2.3 Gambar *low contrast* dan *high contrast* dengan histogramnya.

(Sumber : Gonzalez & Woods, 2002, p. 107)

2.2 Logika Fuzzy

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Ada beberapa alasan mengapa orang menggunakan logika *fuzzy*, antara lain:

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.2.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki dua kemungkinan, yaitu:

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Sedangkan pada himpunan *fuzzy* nilai keanggotaannya terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A[x] = 0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A[x] = 1$ berarti x menjadi anggota penuh pada himpunan A . Himpunan *fuzzy* memiliki 2 atribut, yaitu:

- a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti MUDA, PAROBAYA, TUA.
- b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

- a. Variabel *fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*.
- b. Himpunan *fuzzy*, merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.
- c. Semesta Pembicaraan, adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan.
- d. Domain, merupakan keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*.

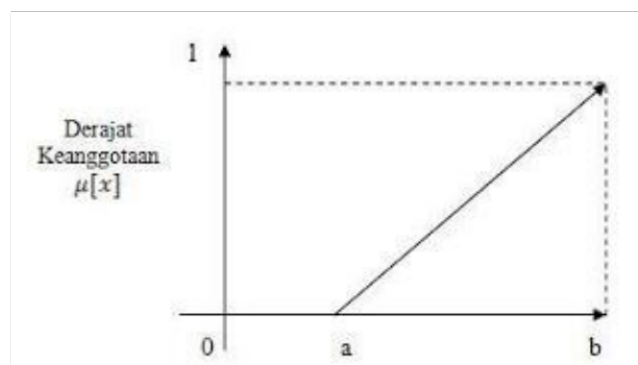
2.2.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Ada beberapa fungsi yang bisa digunakan.

1. Representasi Kurva Linear

Pada representasi kurva linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Terdapat 2 jenis keadaan pada representasi tersebut yakni

- a. Linear naik, himpunan yang dimulai dari domain dengan nilai keanggotaan 0 ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

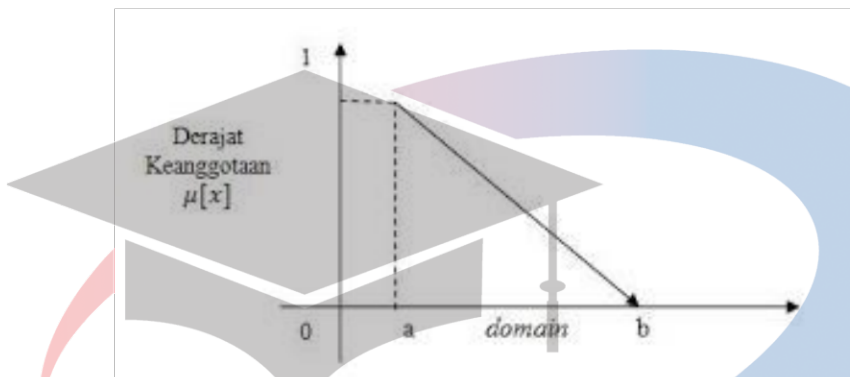


Gambar 2.4 Kurva Linear Naik

Dengan fungsi keanggotaan sebagai berikut.

$$\mu [x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad \dots(2.2)$$

- b. Linear turun, himpunan yang dimulai dari domain dengan nilai keanggotaan 1 ke nilai domain yang memiliki derajat keanggotaan 0.



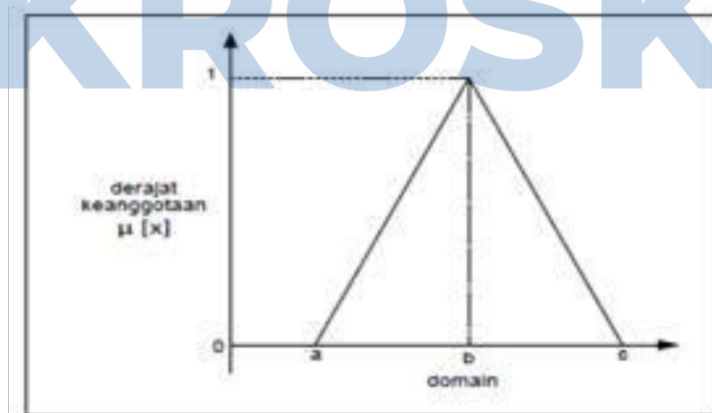
Gambar 2.5 Kurva Linear Turun

Dengan fungsi keanggotaan sebagai berikut.

$$\mu [x] = \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad \dots(2.3)$$

2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti yang terlihat pada gambar 2.6.



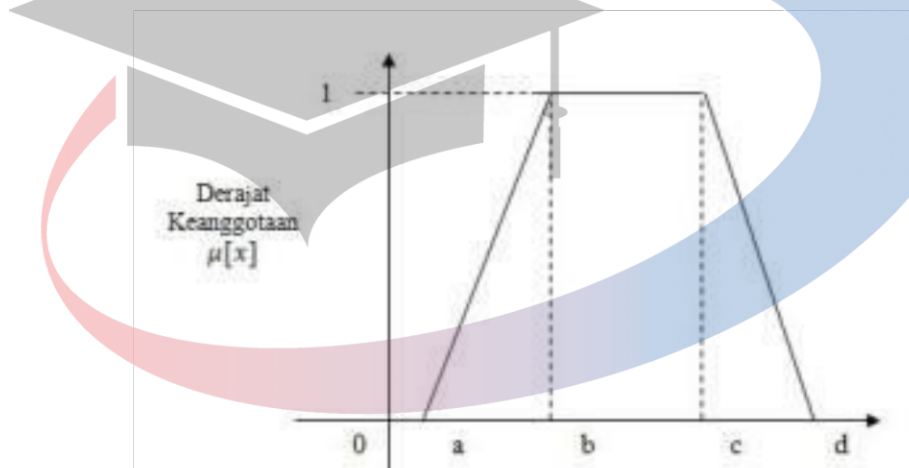
Gambar 2.6 Kurva Segitiga

Dengan fungsi keanggotaan sebagai berikut.

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (b - x)/(c - b); & b \leq x \leq c \end{cases} \quad \dots(2.4)$$

3. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 (gambar 2.7)



Gambar 2.7 Kurva Trapesium

Dengan fungsi keanggotaan sebagai berikut. (Kusumadewi & Purnomo, 2004)

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d - x)/(d - c); & x \geq d \end{cases} \quad \dots(2.5)$$

2.3 Pengolahan Citra

Meskipun sebuah citra kaya informasi, namun seringkali citra mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang

disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*). (Munir, 2004, p. 3).

2.3.1 Konversi Citra Warna ke Grayscale

Konversi citra warna ke citra *grayscale* dapat dilakukan dengan berbagai cara. Beberapa diantaranya yaitu : konversi warna citra ke *grayscale* menggunakan cara klasik dan konversi citra warna ke *grayscale* menggunakan teknik *luma*.

Konversi citra ke *grayscale* dengan menggunakan teknik klasik dilakukan dengan rumus berikut.

$$Gray = (R + G + B) / 3 \quad \dots(2.6)$$

Gray menunjukkan nilai *gray* yang baru, *R* adalah nilai *Red* pada citra warna asli, *G* adalah nilai *Green* pada citra warna asli, dan *B* adalah nilai *Blue* pada citra warna asli.

Pada proses konversi citra warna ke *grayscale* dengan menggunakan algoritma *luma* dapat dilakukan dengan menggunakan rumus berikut. (Tanner, 2011)

$$Gray = (R * 0.3) + (G * 0.59) + (B * 0.11) \quad \dots(2.7)$$

Keunggulan algoritma *luma* terdapat pada penerapan nilai *Green* lebih besar dibandingkan dengan *Red*, dan nilai *Red* lebih besar dibandingkan dengan nilai *Blue*. Penerapan ini didapatkan dari konsep daya tangkap mata manusia terhadap warna. Mata manusia lebih peka dalam menerima warna hijau dibandingkan dengan warna merah, dan lebih peka menerima warna merah dibandingkan warna biru. Dari konsep ini, dibentuk banyak rumus pada algoritma *luma* untuk menghasilkan nilai *gray* dan yang paling umum digunakan adalah rumus 2.7.

2.3.2 Perbaikan Kualitas Citra

Perbaikan citra bertujuan untuk meningkatkan kualitas tampilan citra untuk pandangan manusia atau untuk mengkonversi suatu citra agar memiliki format yang

lebih baik sehingga citra tersebut menjadi lebih mudah diolah dengan mesin (komputer). (Putra, 2010, p. 119).

2.3.3 Peregangan Kontras (*Contrast Stretching*)

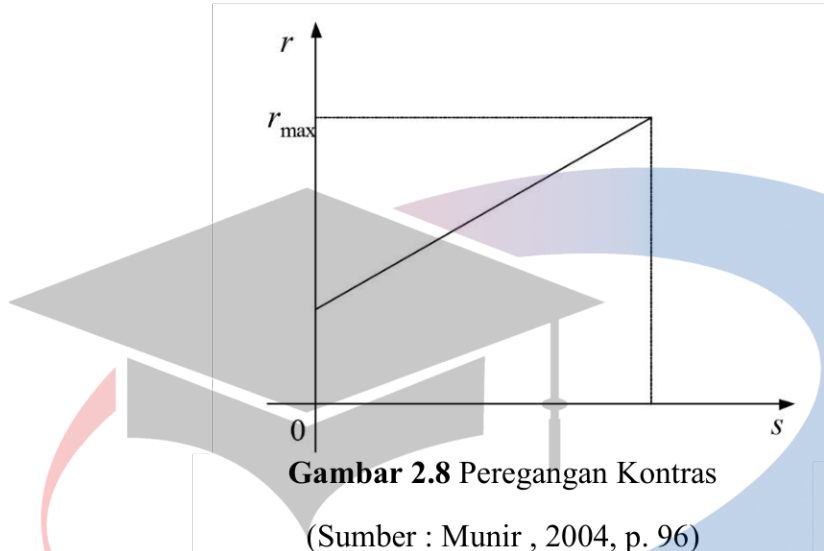
Peregangan kontras adalah teknik yang digunakan untuk memperbaiki kontras citra terutama citra yang memiliki kontras rendah. Melalui operasi ini, nilai-nilai keabuan *pixel* akan merentang dari 0 sampai 255 (pada citra 8-bit), dengan kata lain seluruh nilai keabuan *pixel* terpakai secara merata.

Algoritma peregangan kontras adalah sebagai berikut.

1. Cari batas bawah pengelompokan *pixel* dengan cara memindai (*scan*) histogram dari nilai keabuan terkecil ke nilai keabuan terbesar (0 sampai 255) untuk menemukan *pixel* pertama yang melebihi nilai ambang pertama yang telah dispesifikasikan.
2. Cari batas atas pengelompokan *pixel* dengan cara memindai histogram dari nilai keabuan tertinggi ke nilai keabuan terendah (255 sampai 0) untuk menemukan nilai *pixel* pertama yang lebih kecil dari nilai ambang kedua yang dispesifikasikan.
3. *Pixel-pixel* yang berada dibawah nilai ambang pertama di-*set* sama dengan 0, sedangkan *pixel-pixel* yang berada di atas nilai ambang kedua di-*set* sama dengan 255.
4. *Pixel-pixel* yang berada diantara nilai ambang pertama dan nilai ambang kedua dipetakan (diskalakan) untuk memenuhi rentang nilai-nilai keabuan yang lengkap (0 sampai 255) dengan persamaan:

$$s = \frac{r - r_{\max}}{r_{\min} - r_{\max}} \times 255 \quad \dots\dots\dots(2.8)$$

yang dalam hal ini, r adalah nilai keabuan dalam citra semula, s adalah nilai keabuan yang baru, r_{min} adalah nilai keabuan terendah dari kelompok *pixel*, dan r_{max} adalah nilai keabuan tertinggi dari kelompok *pixel*. (Gambar 2.8) (Munir , 2004, pp. 94-96).

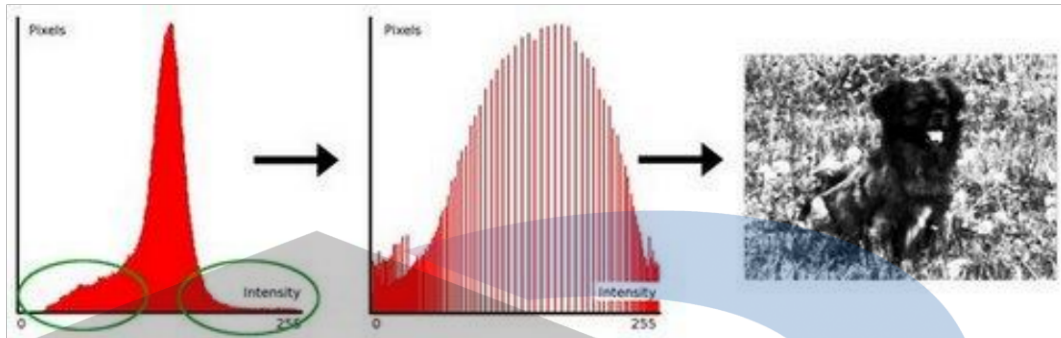


2.3.4 Histogram Equalization

Histogram citra merupakan diagram yang menggambarkan frekuensi setiap nilai intensitas yang muncul di seluruh piksel citra. Nilai besar menyatakan bahwa piksel-piksel yang mempunyai intensitas tersebut sangat banyak. (Kadir & Susanto, 2013, p. 36).

Ekualisasi Histogram adalah suatu proses untuk meratakan histogram agar derajat keabuan dari yang paling rendah (0) sampai dengan yang paling tinggi (255) mempunyai kemunculan yang rata. Dengan *histogram equalization*, hasil gambar yang memiliki histogram yang tidak merata atau distribusi kumulatif yang banyak loncatan gradiasinya akan menjadi gambar yang lebih jelas karena derajat keabuannya tidak dominan gelap atau dominan terang. Proses *histogram equalization* ini menggunakan distribusi kumulatif, karena pada proses ini dilakukan perataan *gradient* dari distribusi kumulatifnya. Tujuan dari HE adalah untuk memperoleh penyebaran histogram yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. (Sutoyo, et al., 2009, p. 46)

Dengan menggunakan *histogram equalization*, maka histogram hasil ekualisasi akan disebarakan (*spreading*). Hasil *histogram equalization* dan histogramnya dapat dilihat pada gambar 2.9



Gambar 2.9 Hasil *histogram equalization* dan histogram

(Sumber : OpenCV Documentation 2.4.11.0)

Langkah awal untuk melakukan *histogram equalization* adalah penentuan *probability density function* (PDF) dari citra dengan menggunakan rumus :

$$p(k) = \frac{H(k)}{N}, \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.9)$$

Dimana N adalah total *pixel* yang berada di dalam citra, $H(k)$ menyatakan nilai intensitas k didalam citra dan L adalah total tingkat keabuan yang terdapat pada citra. Langkah selanjutnya adalah penentuan fungsi kumulatif dengan menggunakan *cumulative density function* (CDF), yang didefinisikan sebagai berikut:

$$c(k) = \sum_{i=0}^k p(i), \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.10)$$

Langkah terakhir adalah melakukan pemetaan tingkat keabuan kembali dengan menggunakan rumus transformasi yang berikut.

$$f(k) = X_0 + (X_{L-1} - X_0) \cdot c(k) \quad \dots(2.11)$$

Dimana X_0 dan X_{L-1} menyatakan tingkat keabuan terendah dan tertinggi secara berurutan. HE melakukan pemetaan ulang citra awal ke seluruh rentang nilai intensitas $[X_0, X_{L-1}]$.

2.4 Adaptive Histogram Equalization (AHE) on Image Gray Level Mapping

AHE on image gray level mapping tersebut memiliki ide yakni f_i adalah nilai keabuan dari tingkat keabuan ke- i yang terdapat pada citra asli. Posisi j dari tingkat keabuan yang telah dipetakan g_j ditentukan dari rasio dari $\sum_{k=0}^{i-1} p_k$ dan $\sum_{k=i+1}^{m-1} p_k$. Untuk mencapai distribusi seragam atau distribusi seragam lokal, algoritma melakukan perbandingan i dengan j : jika $j < i$, maka dilakukan pemetaan secara *ascending*. Jika $j > i$, maka dilakukan pemetaan secara *descending*.

$$j = (m-1) \frac{\sum_{k=0}^{i-1} p_k}{\sum_{k=0}^{i-1} p_k + \sum_{k=i+1}^{m-1} p_k} \quad \dots(2.12)$$

Dimana, $\sum_{k=0}^{m-1} p_k = 1, p_k = \frac{q_k}{Q}$.

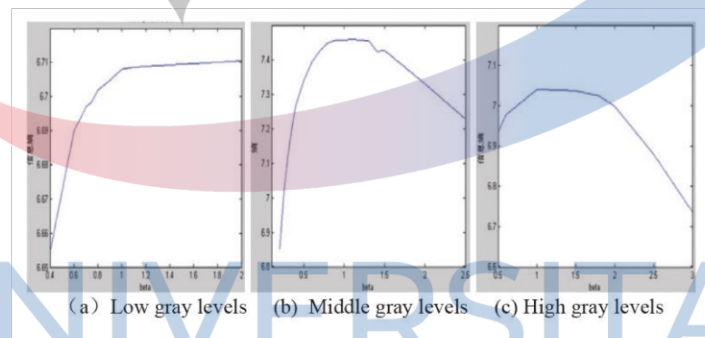
Pada proses pemetaan, tingkat keabuan dengan jumlah frekuensi kemunculan yang kecil akan tertutup oleh tingkat keabuan dengan jumlah frekuensi yang besar. Hal ini yang menyebabkan terjadinya kehilangan informasi. Untuk mencegahnya, maka di perkenalkan sebuah parameter adaptif β di dalam proses pemetaan keabuan. Untuk mendapatkan efek visual yang lebih baik, maka digunakan fungsi *entropy* sebagai fungsi objektif untuk memilih β secara adaptif berdasarkan distribusi keabuan yang terdapat pada citra awal. Hubungan pemetaannya adalah

$$q_k = \log(q_k + 1) \quad \dots(2.13)$$

$$j = (m-1) \frac{\sum_{k=0}^{i-1} p_k}{\sum_{k=0}^{i-1} p_k + \beta \sum_{k=i+1}^{m-1} p_k}, \quad \beta \in (0, +\infty) \quad \dots(2.14)$$

Penyeleksian parameter adaptif β :

Dari rumus (2.14), telah jelas bahwa j merupakan fungsi penurunan monoton dari β . Jika sebuah citra cenderung gelap, maka tingkat keabuan akan berkumpul secara berlebihan di sisi kiri dari histogram. Untuk mendapatkan efek visual yang lebih baik, maka j harus ditambahkan dan β harus lebih kecil dari 1. Pada gambar 2.10(a), nilai β yang tepat adalah 0,8. Jika tingkat kecerahan citra adalah sedang, maka tingkat keabuan berkumpul di bagian tengah dari histogram. Dari gambar 2.10(b), nilai β yang tepat adalah 1,1. Jika sebuah citra cenderung terang, maka tingkat keabuannya berkumpul di sisi kanan dari histogram, nilai j harus dikurangi dan β harus lebih besar dari 1. Dari gambar 2.10(c), nilai β yang tepat adalah 1,5. Berdasarkan pernyataan sebelumnya, maka dapat disimpulkan bahwa nilai β berhubungan dengan tingkat keabuan yang terdapat pada citra awal. Relasi antara *entropy* dengan β di tunjukkan pada gambar 2.6 berikut.



Gambar 2.10 Relasi antara *entropy* dengan β

(Sumber : Zhu & Huang, 2012)

Definisi tingkat keabuan:

Misalkan sebuah citra mempunyai 256 tingkat keabuan. Itu dapat dibagi menjadi 3 jenis : tingkat keabuan rendah (*low gray levels*), tingkat keabuan menengah (*middle gray levels*), dan tingkat keabuan tinggi (*high gray levels*). Ditetapkan *threshold* TL=85, TH = 170. Jika nilai keabuan berada dibawah 85, maka dapat diklasifikasikan sebagai tingkat keabuan rendah; jika nilai keabuan berada diantara 85 dan 170, maka dapat diklasifikasikan sebagai tingkat keabuan menengah; jika nilai keabuan berada diatas 170, maka dapat diklasifikasikan sebagai tingkat keabuan tinggi. Di saat yang bersamaan, jumlah *pixel* untuk tingkat

keabuan rendah, menengah, dan tinggi dihitung masing-masing dan disimpan sebagai *num_low*, *num_mid*, *num_high*. Nilai tertinggi dari ketiganya akan menentukan jenis citra. Jika *num_low* merupakan yang terbesar, maka citra tersebut merupakan citra yang sangat gelap. (Zhu & Huang, 2012)

2.5 Adaptive Contrast Enhancement Algorithm with Details Preserving (ACEDP)

ACEDP memperkenalkan teknik yang telah di modifikasi untuk melakukan peningkatan kontras sebuah citra sambil mempertahankan detail dari citra. ACEDP terdiri dari beberapa langkah yaitu:

1. Klasifikasikan jenis citra berdasarkan jumlah terbanyak dari nilai intensitasnya *pixel*.

Pertama, histogram dari citra awal dibentuk. Dua nilai *threshold* dibentuk, yang dinamakan *upper threshold* dan *lower threshold* dimana nilainya

```
1: IF maximum_no_of_pixels_intensities < 85
2: THEN image_type=low gray level
3: ELSE IF maximum_no_of_pixels_intensities >170
4: THEN image_type=high gray level
5: ELSE image_type=middle gray level
```

Gambar 2.11 Klasifikasi Jenis Citra

(Sumber : Tang & Mat Isa, 2014)

adalah 85 dan 170 secara berurutan. Kedua *threshold* tersebut akan membagi histogram menjadi 3 bagian yang sama besar. (Zhu & Huang, 2012). Citra diklasifikasikan menjadi citra dengan tingkat keabuan rendah, menengah, dan tinggi berdasarkan jumlah terbanyak dari intensitas *pixel*.(Gambar 2. 11).

2. Menentukan *Plateau Levels*

Pada ACEDP telah menetapkan beberapa fungsi untuk *histogram clipping* berdasarkan jenis citranya. Jika citra yang relatif gelap dengan jumlah terbanyak dari intensitas *pixel* lebih kecil dari 85, maka akan menggunakan rumus (2.15). Hal yang sama juga dilakukan untuk citra dengan tingkat keabuan menengah dan tinggi, maka fungsi *plateau level* yang digunakan adalah rumus (2.16) dan (2.17) secara berurutan. (Gambar 2.12). Konstanta c_1 dan c_2 yang digunakan memiliki rentang $[-0.015, -0.005]$ dan $[0.005, 0.007]$ secara berurutan.

1:	IF <i>image_type</i> = <i>low_gray_level</i>	
2:	THEN $level = c_1 + \max(pdf)$	(2.15)
3:	IF <i>image_type</i> = <i>middle_gray_level</i>	
4:	THEN $level = \text{mean}(pdf)$	(2.16)
5:	IF <i>image_type</i> = <i>high_gray_level</i>	
6:	THEN $level = c_2 + \text{mean}(pdf)$	(2.17)

Gambar 2.12 Determinasi dari *plateau level* berdasarkan jenis citra

(Sumber : Tang & Mat Isa, 2014)

3. *Histogram Clipping* dan *Equalization*

Dengan *plateau level* yang didapatkan pada langkah sebelumnya, maka *histogram clipping* dijalankan.

Tentukan terlebih dahulu histogram untuk intensitas k , $P(k)$ dinyatakan dengan rumus berikut:

$$P(k) = n_k, \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.18)$$

Dimana n_k adalah jumlah kemunculan intensitas k di dalam citra dan L adalah total tingkat keabuan yang terdapat pada citra. *Probability density function* (PDF) dari citra, $r(k)$ dinyatakan dengan rumus :

$$r(k) = \frac{P(k)}{N}, \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.19)$$

Dimana N adalah jumlah *pixel* didalam citra. Penjumlahan dari seluruh $r(k)$ sama dengan 1 yang terlihat pada rumus (2.20).

$$\sum_{i=0}^{L-1} r(i) = 1 \quad \dots(2.20)$$

Cumulative density function (CDF), c_k dinyatakan dengan rumus berikut:

$$c(k) = \sum_{i=0}^k r(i), \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.21)$$

Proses *clipping* pada histogram dilakukan dengan menggunakan rumus berikut.

$$P_{clip} = \begin{cases} P(k), & \text{for } P(k) \leq level(k) \\ level(k), & \text{for } P(k) > level(k) \end{cases} \quad \dots(2.22)$$

Setelah proses *clipping*, terapkan fungsi transformasi HE dengan beberapa perubahan seperti yang tertera pada rumus (2.23).

$$f(k) = X_0 + (X_{L-1} - X_0) \cdot \sum_{k=0}^{L-1} P_{clip}(k) \quad \dots(2.23)$$

Dimana X_0 dan X_{L-1} merepresentasikan tingkat keabuan terendah dan tertinggi secara berurutan.

2.6 Adaptive Fuzzy Contrast Enhancement Algorithm with Details Preserving (AFCEDP)

AFCEDP mengintegrasikan teknik histogram clipping sebelum melakukan perataan histogram. Dalam penentuan fungsi *clipping limit*, digunakan element *fuzzy* untuk menentukan kategori kontras citra. Ini disebabkan perlakuan untuk tiap fungsi *plateau level* pada tiap kategori kontras citra berbeda-beda. Langkah-langkah yang terdapat pada metode ini adalah sebagai berikut

1. Penentuan fungsi keanggotaan untuk tiap intensitas.

Fungsi keanggotaan yang digunakan pada metode ini adalah fungsi keanggotaan trapesium yang dikategorikan untuk 3 kategori kontras citra,

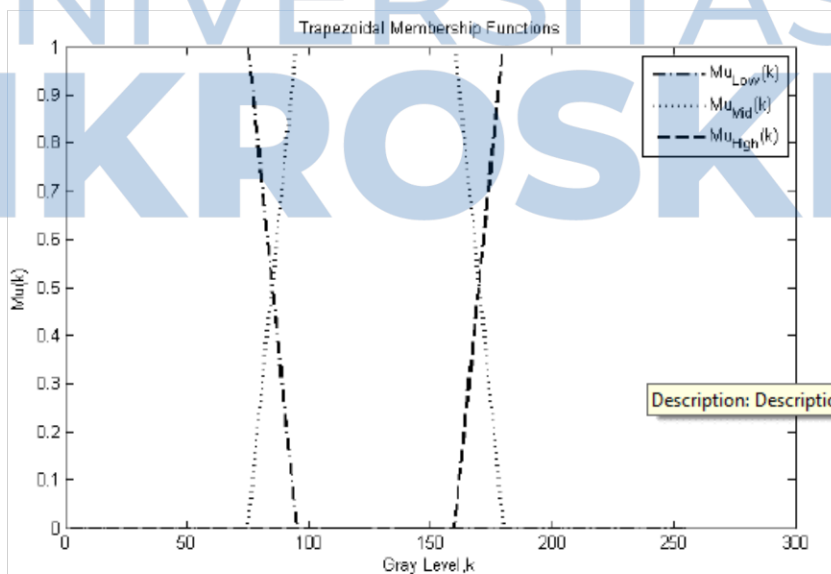
yakni rendah, sedang, dan tinggi (*low*, *mid*, dan *high*) yang didefinisikan sebagai berikut.

$$\mu_{low}(k) = \begin{cases} 0 & , k > 95 \\ \frac{95-k}{20} & , 75 \leq k \leq 95 \\ 1 & , k < 75 \end{cases} \quad \dots (2.24)$$

$$\mu_{mid}(k) = \begin{cases} 0 & , (k < 75) \cup (k > 180) \\ \frac{k-75}{20} & , 75 \leq k \leq 95 \\ 1 & , 95 \leq k \leq 160 \\ \frac{180-k}{20} & , 160 \leq k \leq 180 \end{cases} \quad \dots (2.25)$$

$$\mu_{high}(k) = \begin{cases} 0 & , k < 160 \\ \frac{k-160}{20} & , 160 \leq k \leq 180 \\ 1 & , k > 180 \end{cases} \quad \dots (2.26)$$

dimana *k* merupakan intensitas pixel pada citra. Bentuk distribusi keanggotaan pada fungsi keanggotaan terdapat pada Gambar 2.13



Gambar 2.13 Distribusi derajat keanggotaan berbentuk trapesium

(Sumber : Tang & Mat Isa, 2014)

- Perhitungan derajat keanggotaan dan nilai intensitas referensi

Untuk mendapatkan nilai intensitas referensi dari fungsi keanggotaan trapesium, derajat dari citra yang termasuk diantara tiga kategori menggunakan partisi dihitung. Nilai intensitas referensi akan digunakan pada tahapan selanjutnya. Perhitungan intensitas referensi dilakukan sebagai berikut.

$$\lambda = (\text{low_part} \times 43) + (\text{mid_part} \times 128) + (\text{high_part} \times 213) \dots (2.27)$$

- Mendefinisikan 3 Fungsi *Plateau* dan melakukan komputasi *Clipping Limit*.

Teknik AFCEDP menggunakan fungsi *clipping* yang sama dengan teknik ACEDP. Sesuai yang dijelaskan di ACEDP, jarak yang diterima untuk *slopes c1* dan *c2* adalah [-0.015,-0.005] dan [0.005,0.007] secara berurutan. Pada AFCEDP menggunakan nilai *c1* dan *c2* yang sama dengan ACEDP. Nilai untuk *c1* dan *c2* yang digunakan adalah -0.01 dan 0.007 secara berurutan. Perhitungan untuk fungsi *clipping*, $\sigma(k)$, sebagai berikut.

$$\sigma(k) = [\mu_{\text{low}}(\lambda) \times \text{level}_{\text{low}}(k)] + [\mu_{\text{mid}}(\lambda) \times \text{level}_{\text{mid}}(k)] + [\mu_{\text{high}}(\lambda) \times \text{level}_{\text{high}}(k)] \dots (2.28)$$

Fungsi *plateau* yang diterapkan adalah sebagai berikut.

$$\text{level}_{\text{low}} = c_1 + \max(\text{pdf})$$

$$\text{level}_{\text{mid}} = \text{mean}(\text{pdf})$$

$$\text{level}_{\text{high}} = c_2 + \text{mean}(\text{pdf})$$

Gambar 2.14 Fungsi *plateau*

(Sumber : Tang & Mat Isa, 2014)

- Lakukan *Clipping* dan Ekualisasi *histogram*.

Fungsi *clipping* $\sigma(k)$ menyediakan pembatasan untuk tiap tingkat keabuan. Anggap bentuk citra masukkan berupa citra *grayscale*, maka *histogram* dari citra, $H(k)$ didefinisikan sebagai berikut.

$$H(k) = n_k, \text{ for } k = 0, 1, \dots, L-1 \dots (2.29)$$

Dimana n_k adalah jumlah kemunculan dari intensitas k di citra dan L adalah total tingkat keabuan didalam citra. Fungsi probabilitas densitas dari citra didefinisikan sebagai berikut.

$$p(k) = \frac{H(k)}{N}, \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.30)$$

Dimana N adalah total piksel didalam citra.

Fungsi kumulatif densitas, $c(k)$ didefinisikan sebagai berikut.

$$c(k) = \sum_{i=0}^k p(i), \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.31)$$

HE menggunakan fungsi transformasi untuk memetakan tingkat keabuan masukan menjadi tingkat keabuan yang baru, fungsi transformasi $f(k)$ didefinisikan sebagai berikut.

$$f(k) = X_0 + (X_{L-1} - X_0) \cdot c(k) \quad \dots(2.32)$$

dimana X_0 dan X_{L-1} merepresentasikan batas bawah dan batas atas dari histogram secara berurutan.

Fungsi transformasi baru $new_f(k)$, menawarkan peningkatan ketajaman citra sesuai dengan rumus (2.33) berikut. Fungsi tersebut akan menggantikan fungsi transformasi sebelumnya. (Ooi, et al., 2009).

$$new_f(k) = X_0 + (X_{L-1} - X_0) \cdot (c(k) - \frac{1}{2} p(k)) \quad \dots(2.33)$$

Beberapa fungsi baru lain yang diterapkan di algoritma tersebut adalah fungsi probabilitas densitas dan fungsi kumulatif densitas yang terlihat pada rumus (2.34) dan (2.35) secara berurutan.

$$new_p(k) = \min(p(k), \sigma(k)), \text{ for } k = 0, L-1 \quad \dots(2.34)$$

$$c(k) = \sum_{i=0}^k new_p(i), \text{ for } k = 0, 1, \dots, L-1 \quad \dots(2.35)$$

Lakukan pencarian nilai $p(k)$ baru dengan rumus (2.34), setelah itu lakukan perhitungan $c(k)$ yang baru dengan rumus (2.35).

Karena telah dilakukan *clipping* maka nilai kumulatif dari pdf tidak akan menjadi 1. Untuk mendapatkan nilai kumulatif 1, maka dilakukan rumus dibawah ini untuk mendapatkan nilai kumulatif = 1.

$$new_p(k)' = new_p(k) / \sum pdf \quad \dots(2.36)$$

Setelah mendapatkan nilai pdf baru maka lakukan kembali rumus (2.35) dengan menggantikan nilai $new_p(k)$ dengan nilai $new_p(k)'$ yang didapat dari rumus (2.36) untuk mendapatkan nilai $new_c(k)$, Langkah terakhir adalah lakukan fungsi transformasi dengan menggunakan fungsi transformasi baru pada rumus (2.35), dengan nilai $new_p(k)'$ yang baru hasil dari rumus (2.36).

2.7 Perbandingan Citra

Perbandingan citra digunakan untuk membandingkan kemiripan antar citra secara matematis. Metode yang digunakan untuk melakukan perbandingan citra adalah *Shannon Entropy* dan *Contrast Improvement Evaluation*.

2.7.1 Shannon Entropy dan Details Preserving

Tujuan utama dari *Histogram Equalization* (HE) adalah menghasilkan suatu citra dengan probabilitas kemunculan yang seragam pada tiap nilai intensitas keabuannya. Hal ini dapat dijelaskan dengan salah satu sifat dari *Shannon Entropy* (Shannon, 1948), dimana citra dinyatakan memiliki kualitas terbaik apabila probabilitas kemunculan pada tiap intensitas keabuan seragam. (Tang & Mat Isa, 2014). *Details Preserving* bertujuan untuk melestarikan informasi dari citra awal sehingga tidak hilang pada saat HE dilakukan.

Shannon Entropy merupakan rumus matematika yang secara luas digunakan untuk menghitung kekayaan informasi. Semakin tinggi nilai *Entropy* maka semakin tinggi pula detail dan informasi yang dimiliki oleh citra tersebut. Rumus *Shannon Entropy* dinyatakan pada rumus (2.37).

$$E = -\sum_{i=0}^N r(i) \log_2 r(i) \quad \dots(2.37)$$

Dimana $r(i)$ merupakan probabilitas kemunculan nilai keabuan, N adalah nilai keabuan tertinggi.

Nilai *Entropy* tertinggi terdapat pada citra yang memiliki nilai histogram untuk tiap keabuan dengan nilai yang sama, dan memiliki seluruh intensitas keabuan dari nilai 0 hingga 255. Nilai *Entropy* dapat mewakili seberapa besar *details preserving* yang terjadi pada saat dilakukan proses peningkatan kontras citra.

Awalnya nilai *Entropy* untuk citra awal dan citra hasil dihitung, kemudian akan dilakukan perbandingan dengan cara membagikan nilai *Entropy* citra hasil dengan citra awal untuk mencari persentase pelestarian detail yang terjadi pada proses peningkatan kontras tersebut dan seberapa besar informasi yang hilang.

2.7.2 Contrast Improvement Evaluation

Untuk mengetahui perbedaan atau peningkatan nilai kontras pada dua citra yang sama, maka rumus *Contrast Improvement Evaluation* sering dimanfaatkan untuk alat ukur peningkatan kontras pada dua citra yang sama. Rumus *Contrast Improvement Evaluation* dapat dilihat pada rumus (2.38).

$$C = 10 \log_{10} \left[\frac{1}{WH} \sum_{u=1}^W \sum_{v=1}^H g^2(u,v) - \left| \frac{1}{WH} \sum_{u=1}^W \sum_{v=1}^H g(u,v) \right|^2 \right] \dots(2.38)$$

Dimana W dan H adalah *width* dan *height* (panjang dan tinggi) dari citra, $g(u,v)$ adalah intensitas dari piksel di posisi 2 dimensi (u,v) . Semakin besar nilai *Contrast Improvement Evaluation* untuk citra hasil, maka berarti semakin bagus juga peningkatan kontras yang terjadi. (Tang & Mat Isa, 2014).